Healthcare Centers Database System:
Design and Implementation

INTRODUCTION

There are many types of Healthcare Facilities: hospitals, nursing homes, healthcare centers such as doctor's office and clinics, urgent care centers and more. Each one of these facilities requires a specific database designed and crafted to the needs of the facility, its workers and its patients. We will go over the creation and implementation of one such database. Our database is particularly a collection of small sized healthcare centers. These centers would be simple health clinics, also known as doctor's offices.

DATABASE REQUIREMENTS

For the purpose of this project we will pretend that the database was contracted to us by a wealthy businessman owning several different chains of health clinics. Based upon our talks we have gathered the following requirements. The database will deal with few different franchises of health clinics. Each franchise will possibly have multiple branches. A list of employees will be required but only doctors and nurses will be interacting with entities within the database. For this we will need to create two specializations for the Employee entity. Each employee will be allowed to work at one Branch only. The database will have to record rooms existing at the branches as each appointment will be tied to them as well as a patient and the doctor plus the nurse overseeing the patient. Since prescriptions and vaccines will be provided we will need separate entities for them, tied to its provider and receiver. Each patient should have insurance but if they do not have it they will be listed as "selfpay". Not all existing insurances will be accepted so a list of accepted insurances will be kept. Health history will be kept for all patients in the database. Lastly, patients will have bills assigned to them with the status of whether they have been paid or not.

DATABASE CONSTRAINTS - Cardinality

Health_Clinic – Branch: 1:N
Each Branch can belong only to one clinic and a clinic can have many branches.

Employee – Branch: N:1
Employee can work at only one branch while many employees can work at a branch.

Branch – Room: 1:N
A branch can have many rooms but rooms are tied only to one branch.

Appointment – Nurse, Doctor, Room, Patient: 1:1
Each appointment will be between only 1 doctor, 1 nurse, 1 patient and within 1 room.

Doctor - Prescription: 1:N, Prescription – Patient: N:1

Each prescription will be unique and tied only to 1 doctor and 1 patient but a doctor can prescribe many prescriptions and a patient can be prescribed many prescriptions.

Nurse – Vaccine: 1:N, Vaccine – Patient: N:1
Like with prescriptions, each vaccine will be unique and tied to only 1 nurse and 1 patient but a nurse can give many vaccine shots and a patient can get multiple vaccines.

Patient – Bill: 1:N
A patient can receive many bills but each bill will be unique and assigned to one patient only.

Patient – Insurance: N:1
A patient can have only 1 insurance while 1 insurance can belong to many patients

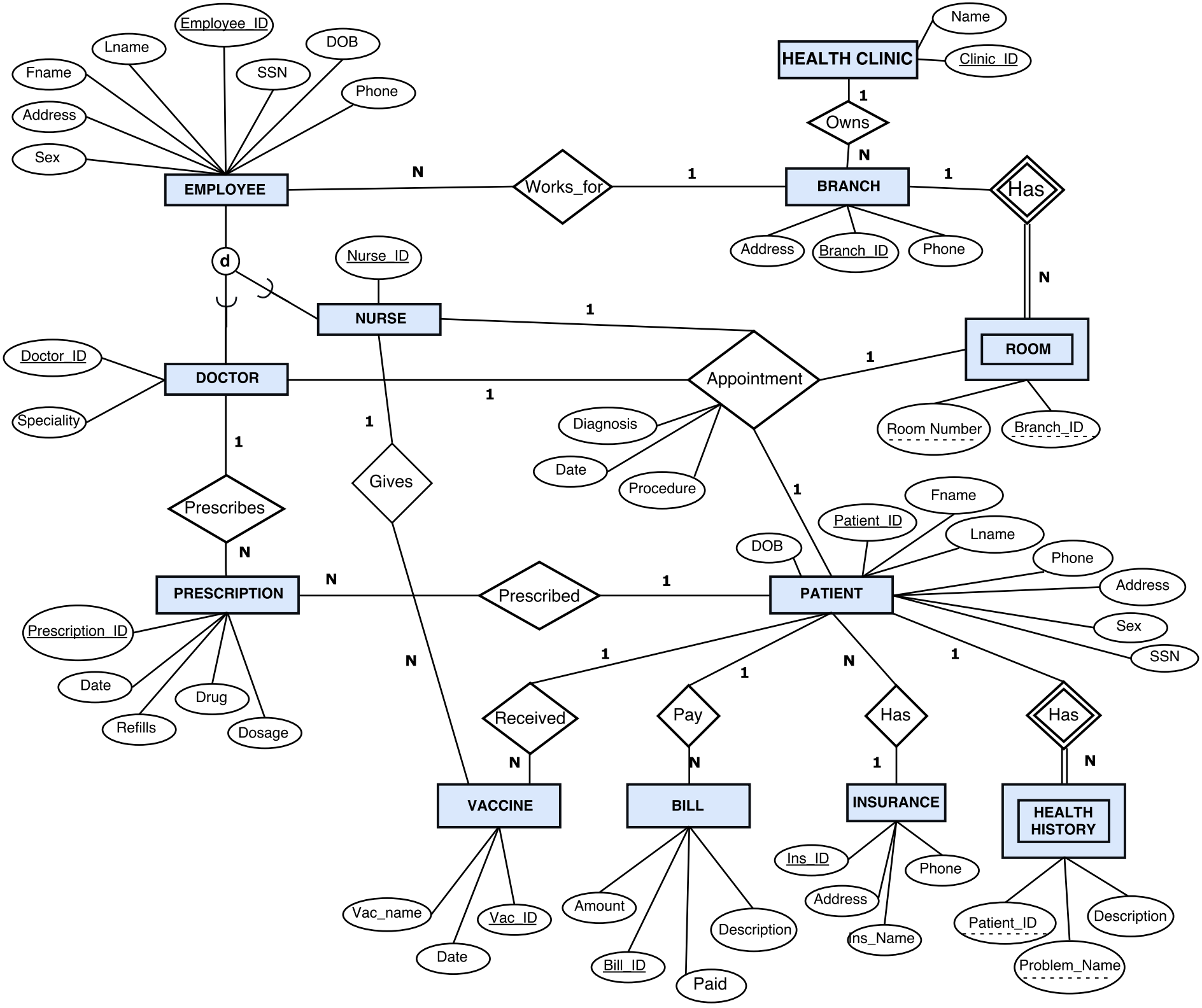
Patient – Health History: 1:N
A health history will be assigned to a single patient and a patient can have many entries in their health history

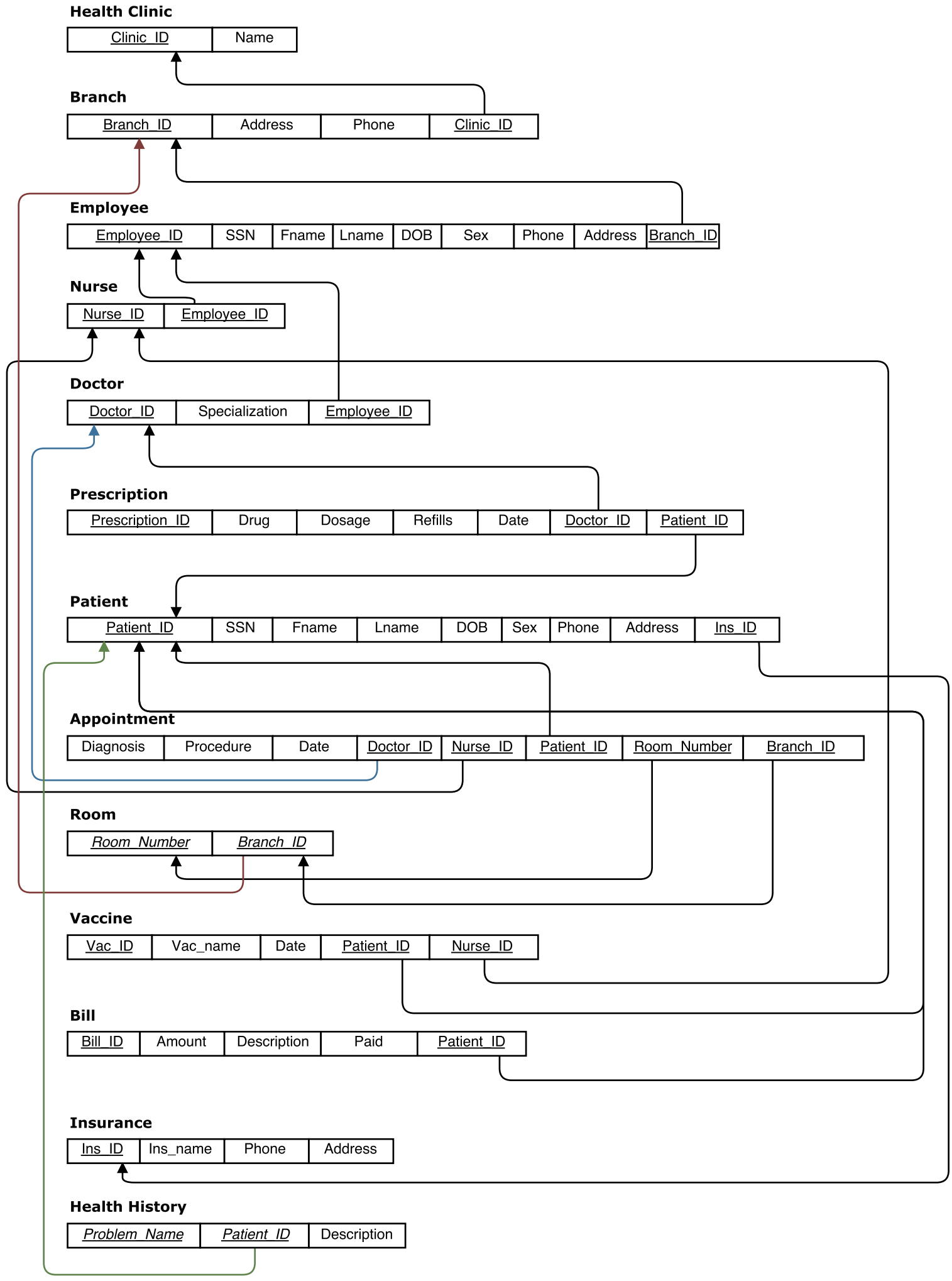
INTEGRITY CONSTRAINTS – Data Types

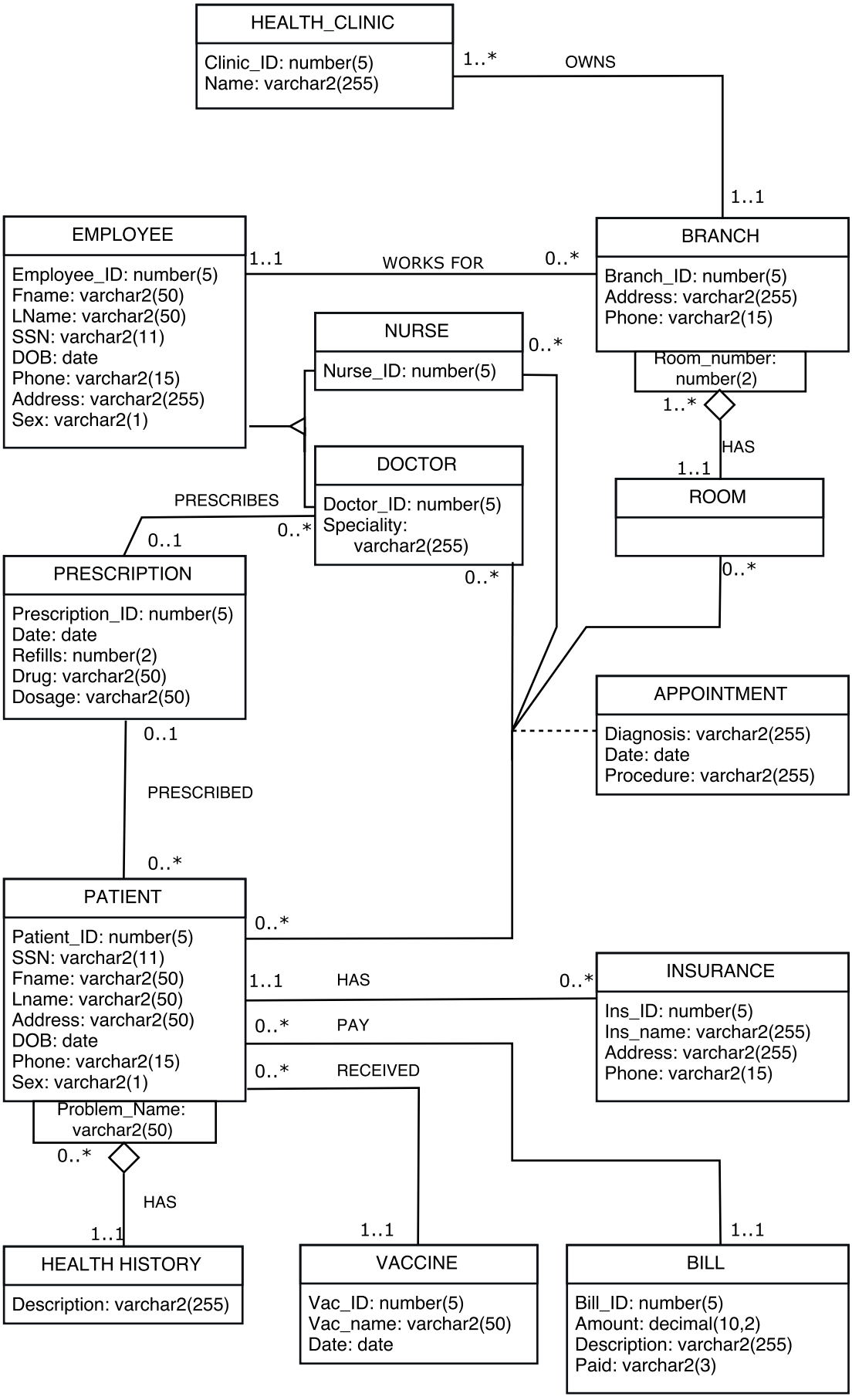
Each attribute with following names will have these data constraints:

- Phone: varchar2(15) {-,0...9}
- SSN: varchar2(11) {-,0...9}
- Sex: varchar2(1) {M,F}
- DOB: date
- Date: date

KEY CONSTRAINTS - Primary Key		**Foreign Key**
Health Clinic –	Clinic_ID	----
Branch –	Branch_ID	Clinic_ID
Employee –	Employee_ID	Branch_ID
Doctor –	Doctor_ID	Employee_ID
Nurse –	Nurse_ID	Employee_ID
Prescription –	Prescription_ID	Doctor_ID, Patient_ID
Patient –	Patient_ID	Ins_ID
Vaccine –	Vac_ID	Patient_ID, Nurse_ID
Bill –	Bill_ID	Patient_ID
Insurance –	Ins_ID	----
Room –	(Room_Number, Branch_ID)	Branch_ID
Health History –	(Problem_Name, Patient ID)	Patient_ID
Appointment(Relationship*)	----	Branch_ID, Nurse_ID, Doctor_ID, Patient_ID


Employee_ID
Lname
DOB
Fname
SSN
Phone
Address
Sex
EMPLOYEE
N
Works_for
1
BRANCH
1
Has
N
Name
HEALTH CLINIC
Clinic_ID
1
Owns
N
Address
Branch_ID
Phone
d
Nurse_ID
NURSE
1
ROOM
Doctor_ID
DOCTOR
Appointment
1
1
Speciality
1
1
Diagnosis
Room Number
Branch_ID
Date
Procedure
Gives
Prescribes
1
Fname
Patient_ID
Lname
DOB
Phone
Address
N
PRESCRIPTION
N
Prescribed
1
PATIENT
Sex
SSN
Prescription_ID
Date
Drug
Refills
Dosage
N
1
1
N
1
Received
Pay
Has
Has
N
N
1
N
VACCINE
BILL
INSURANCE
HEALTH HISTORY
Ins_ID
Phone
Address
Ins_Name
Vac_name
Vac_ID
Amount
Description
Patient_ID
Description
Date
Bill_ID
Paid
Problem_Name


Health Clinic
Clinic_ID
Name
Branch
Branch_ID
Address
Phone
Clinic_ID
Employee
Employee_ID
SSN
Fname
Lname
DOB
Sex
Phone
Address
Branch_ID
Nurse
Nurse_ID
Employee_ID
Doctor
Doctor_ID
Specialization
Employee_ID
Prescription
Prescription_ID
Drug
Dosage
Refills
Date
Doctor_ID
Patient_ID
Patient
Patient_ID
SSN
Fname
Lname
DOB
Sex
Phone
Address
Ins_ID
Appointment
Diagnosis
Procedure
Date
Doctor_ID
Nurse_ID
Patient_ID
Room_Number
Branch_ID
Room
Room_Number
Branch_ID
Vaccine
Vac_ID
Vac_name
Date
Patient_ID
Nurse_ID
Bill
Bill_ID
Amount
Description
Paid
Patient_ID
Insurance
Ins_ID
Ins_name
Phone
Address
Health History
Problem_Name
Patient_ID
Description


HEALTH_CLINIC
Clinic_ID: number(5)
Name: varchar2(255)
1..*
OWNS
1..1
EMPLOYEE
Employee_ID: number(5)
Fname: varchar2(50)
LName: varchar2(50)
SSN: varchar2(11)
DOB: date
Phone: varchar2(15)
Address: varchar2(255)
Sex: varchar2(1)
1..1
WORKS FOR
0..*
BRANCH
Branch_ID: number(5)
Address: varchar2(255)
Phone: varchar2(15)
NURSE
Nurse_ID: number(5)
0..*
Room_number: number(2)
1..*
HAS
1..1
ROOM
DOCTOR
Doctor_ID: number(5)
Speciality: varchar2(255)
PRESCRIBES
0..*
0..1
0..*
PRESCRIPTION
Prescription_ID: number(5)
Date: date
Refills: number(2)
Drug: varchar2(50)
Dosage: varchar2(50)
0..*
APPOINTMENT
Diagnosis: varchar2(255)
Date: date
Procedure: varchar2(255)
0..1
PRESCRIBED
0..*
PATIENT
Patient_ID: number(5)
SSN: varchar2(11)
Fname: varchar2(50)
Lname: varchar2(50)
Address: varchar2(50)
DOB: date
Phone: varchar2(15)
Sex: varchar2(1)
0..*
1..1
HAS
0..*
INSURANCE
Ins_ID: number(5)
Ins_name: varchar2(255)
Address: varchar2(255)
Phone: varchar2(15)
0..*
PAY
0..*
RECEIVED
Problem_Name: varchar2(50)
0..*
HAS
1..1
1..1
1..1
HEALTH HISTORY
Description: varchar2(255)
VACCINE
Vac_ID: number(5)
Vac_name: varchar2(50)
Date: date
BILL
Bill_ID: number(5)
Amount: decimal(10,2)
Description: varchar2(255)
Paid: varchar2(3)
Data Constraints
Phone: varchar2(15) {-,0...9}
SSN: varchar2(11) {-,0...9}
Sex: varchar2(1) {M,F}
Data types are using SQL plus format.
Note: in UML diagrams the locations for (min max) are reversed compared to the locations used in an ER diagram

ER DIAGRAM, UML DIAGRAM, RELATIONAL SCHEMA:
TABLE CREATION IN SQL PLUS

```
CREATE TABLE  Health_Clinic
(Clinic_ID NUMBER(5) NOT NULL,
Name VARCHAR2(255) NOT NULL,
PRIMARY KEY (Clinic_ID))
tablespace project;

CREATE TABLE Branch
(Branch_ID NUMBER(5) NOT NULL,
Address VARCHAR2(255) NOT NULL,
Phone VARCHAR2(15) NOT NULL,
Clinic_ID NUMBER(5),
PRIMARY KEY (Branch_ID),
FOREIGN KEY (Clinic_ID) REFERENCES Health_Clinic (Clinic_ID))
tablespace project;

CREATE TABLE  Employee
(Employee_ID NUMBER (5) NOT NULL,
Fname VARCHAR2(50) NOT NULL,
Lname VARCHAR2(50) NOT NULL,
SSN VARCHAR2(11) NOT NULL,
DOB DATE NOT NULL,
Phone VARCHAR2(15) NOT NULL,
Address VARCHAR2(255) NOT NULL,
Sex VARCHAR2(1) NOT NULL,
Branch_ID NUMBER(5),
PRIMARY KEY (Employee_ID),
FOREIGN KEY (Branch_ID) REFERENCES Branch (Branch_ID))
tablespace project;

CREATE TABLE Insurance
(Ins_ID NUMBER(5) NOT NULL,
Ins_name VARCHAR2(255) NOT NULL,
Phone NUMBER(15),
Address VARCHAR2(255),
PRIMARY KEY (Ins_ID))
tablespace project;

CREATE TABLE Room
(Room_Number NUMBER(2) NOT NULL,
Branch_ID NUMBER(5) NOT NULL,
PRIMARY KEY (Room_Number, Branch_ID),
FOREIGN KEY (Branch_ID) REFERENCES Branch (Branch_ID))
tablespace project;
```

```
CREATE TABLE Nurse
(Nurse_ID NUMBER(5) NOT NULL,
Employee_ID NUMBER(5),
PRIMARY KEY (Nurse_ID),
FOREIGN KEY (Employee_ID) REFERENCES Employee (Employee_ID))
tablespace project;

CREATE TABLE Doctor
(Doctor_ID NUMBER(5) NOT NULL,
Speciality VARCHAR2(255),
Employee_ID NUMBER(5),
PRIMARY KEY (Doctor_ID),
FOREIGN KEY (Employee_ID) REFERENCES Employee (Employee_ID))
tablespace project;

CREATE TABLE Patient
(Patient_ID NUMBER(5) NOT NULL,
SSN VARCHAR2(11) NOT NULL,
Fname VARCHAR2(50) NOT NULL,
Lname VARCHAR2(50) NOT NULL,
DOB date NOT NULL,
Sex VARCHAR2(1) NOT NULL,
Phone VARCHAR2(15) NOT NULL,
Address VARCHAR2(255) NOT NULL,
Ins_ID NUMBER(5),
PRIMARY KEY (Patient_ID),
FOREIGN KEY (Ins_ID) REFERENCES Insurance (Ins_ID))
tablespace project;

CREATE TABLE Prescription
(Prescription_ID NUMBER(5) NOT NULL,
PDate date NOT NULL,
Drug VARCHAR2(50) NOT NULL,
Dosage VARCHAR2(50) NOT NULL,
Refills NUMBER(2) NOT NULL,
Patient_ID NUMBER(5),
Doctor_ID NUMBER(5),
PRIMARY KEY (Prescription_ID),
FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID),
FOREIGN KEY (Doctor_ID) REFERENCES Doctor (Doctor_ID))
tablespace project;
```

```
CREATE TABLE Vaccine
(Vac_ID NUMBER(5) NOT NULL,
Vac_name VARCHAR2(50) NOT NULL,
VDate date NOT NULL,
Patient_ID NUMBER(5),
Nurse_ID NUMBER(5),
PRIMARY KEY (Vac_ID),
FOREIGN KEY (Nurse_ID) REFERENCES Nurse (Nurse_ID),
FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID))
tablespace project;

CREATE TABLE Bill
(Bill_ID NUMBER(5) NOT NULL,
Amount DECIMAL(10,2) NOT NULL,
Description VARCHAR2(255),
Paid VARCHAR2(3),
Patient_ID NUMBER(5),
PRIMARY KEY (Bill_ID),
FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID))
tablespace project;

CREATE TABLE Appointment
(Diagnosis VARCHAR2(255),
Procedure VARCHAR2(255),
ADate date NOT NULL,
Doctor_ID NUMBER(5),
Nurse_ID NUMBER(5),
Patient_ID NUMBER(5),
Room_Number NUMBER(2),
Branch_ID NUMBER(5),
FOREIGN KEY (Doctor_ID) REFERENCES Doctor (Doctor_ID),
FOREIGN KEY (Nurse_ID) REFERENCES Nurse (Nurse_ID),
FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID),
FOREIGN KEY (Room_Number, Branch_ID) REFERENCES Room (Room_Number, Branch_ID))
tablespace project;

CREATE TABLE Health_History
(Problem_Name VARCHAR2(50) NOT NULL,
Description VARCHAR2(255) NOT NULL,
Patient_ID NUMBER(5),
PRIMARY KEY (Problem_Name, Patient_ID),
FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID))
tablespace project;
```

FORMATING

Since sqlplus on our terminals outputs after data a number of spaces corresponding to the max allowed input length, we have come up with formatting for the output view to make everything easier to look at.
These are the commands that could be run before looking at the database:

```
Set linesize 200;
Set pagesize 50;

column Name format a25 WRAP;
column Fname format a10;
column Lname format a15;
column Address format a25 WRAP;
column Sex format a3;
column Ins_name format a25 WRAP;
column Speciality format a15 WRAP;
column Drug format a20;
column Dosage format a6;
column Vac_name format a10 WRAP;
column Description format a25 WRAP;
column Diagnosis format a25 WRAP;
column Procedure format a20 WRAP;
```

PREDEFINED TRANSACTIONS

- **To view all the tables in the database project:**

SELECT table_name FROM all_tables WHERE tablespace_name='PROJECT';

- **To view prescriptions and to whom they are prescribed to:**

SELECT Fname, Lname, Drug, Dosage, Refills FROM Patient P INNER JOIN Prescription R ON P.patient_ID=R.patient_ID;

- **To view prescriptions, to whom they are prescribed and who prescribed them:**

SELECT P.Fname, P.Lname, Drug, Dosage, Refills, E.Fname AS DoctorFName, E.Lname AS DoctorLname FROM Patient P INNER JOIN Prescription R ON P.patient_ID=R.patient_ID INNER JOIN Doctor D ON R.doctor_ID=D.doctor_ID INNER JOIN employee E ON D.employee_ID=E.employee_ID ORDER BY P.Fname, P.Lname;

- **To view what insurance patients have:**

SELECT Fname, Lname, Ins_name FROM Patient P INNER JOIN Insurance N ON P.Ins_ID=N.Ins_ID;

- **To view what vaccines patients receives:**

SELECT Fname, Lname, Vac_name, vdate, Nurse_ID FROM Patient P INNER JOIN Vaccine V ON P.patient_ID=V.Patient_ID ORDER BY Fname, Lname;

- **To view all patients health histories:**

SELECT Fname, Lname, Problem_Name, Description FROM Patient P INNER JOIN Health_History H ON P.Patient_ID=H.Patient_ID;

- **To view a specific patients health history:**

SELECT Problem_Name, Description FROM Patient P INNER JOIN Health_History H ON P.Patient_ID=H.Patient_ID WHERE P.Patient_ID=;

- **To view all patients appointments:**

SELECT Fname, Lname, Diagnosis, Procedure, Adate, Room_Number, branch_ID FROM appointment A INNER JOIN patient P ON A.patient_ID=P.patient_ID;

- **To view a specific patients appointments:**

SELECT Diagnosis, Procedure, Adate, Room_Number, branch_ID FROM appointment A INNER JOIN patient P ON A.patient_ID=P.patient_ID WHERE P.patient_ID=;

- **To view health clinics and what branches they own:**

SELECT Name, Branch_ID, Phone, Address FROM Health_Clinic H INNER JOIN Branch B ON H.Clinic_ID=B.Clinic_ID ORDER BY Name;

- **To list employees that are doctors and the branch they work at:**

SELECT Fname, Lname, Branch_ID FROM doctor D INNER JOIN employee E ON D.Employee_ID=E.Employee_ID;

- **To list employees that are doctors and the clinic and branch address that they work at:**

SELECT Fname, Lname, Name AS Clinic, B.Address AS Location FROM doctor D INNER JOIN employee E ON D.Employee_ID=E.Employee_ID INNER JOIN Branch B ON E.Branch_ID=B.Branch_ID INNER JOIN Health_Clinic H ON B.Clinic_ID=H.Clinic_ID;